

TP1 - Application de Réalité Augmentée simple

Estimation de pose basée marqueur

1 Introduction

L'objectif de ce premier TP est de créer une application de Réalité Augmentée simple faisant apparaître un vaisseau spatial dans un flux vidéo acquis par une webcam (Figure 1). Pour cela, nous utiliserons un marqueur planaire aux dimensions connues afin d'estimer la pose de la caméra dans le monde réel et ainsi être en mesure de positionner la caméra virtuelle pour générer le rendu du vaisseau spatial. L'application sera programmée en C++ à l'aide des modules de la bibliothèque ViSP¹.

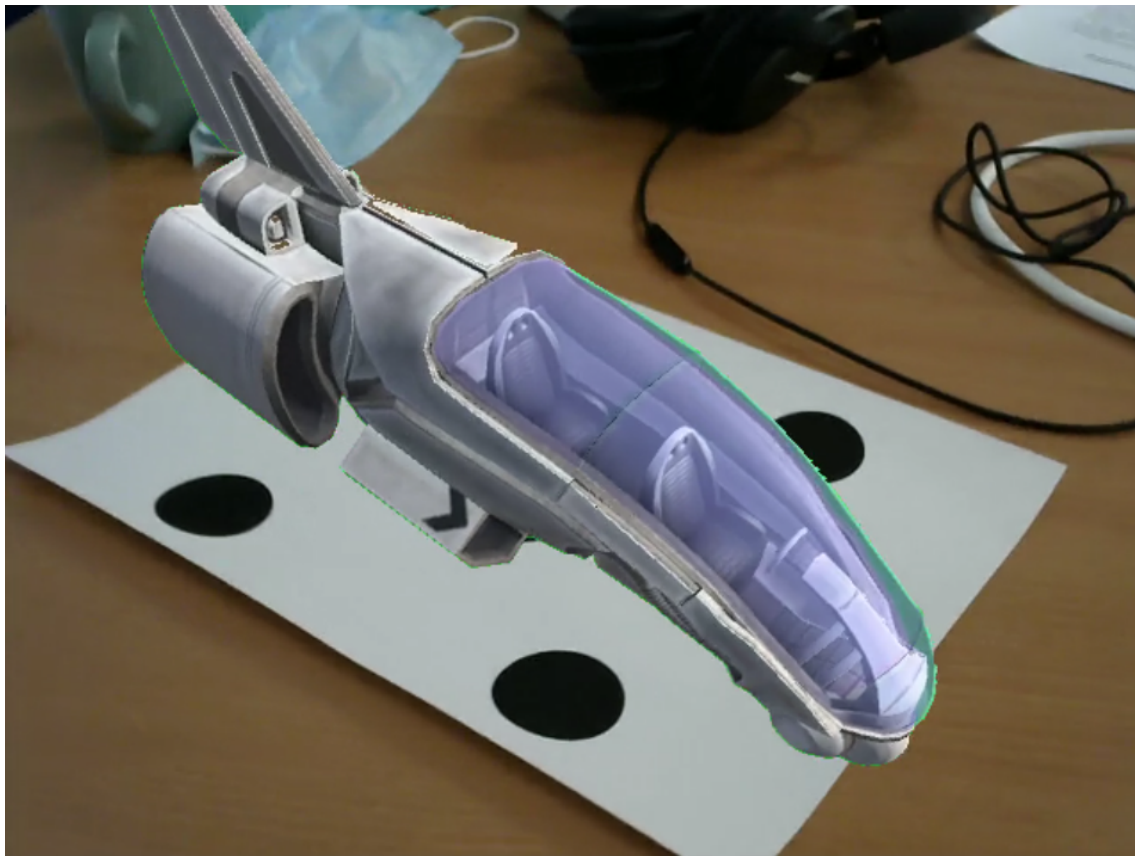


FIGURE 1 – Résultat attendu à l'issu de ce TP

1. <https://visp.inria.fr/>

2 Éléments fournis

Vous trouverez dans le dossier "MV54" fourni :

- Un Standalone Unity "AR_Engine". La scène de cette application Unity ne contient qu'une caméra virtuelle et le modèle 3D du vaisseau spatial sur un fond purement vert (Figure 2). L'application contient un serveur http afin de rendre la caméra virtuelle contrôlable (changement des paramètres extrinsèques/intrinsèques, acquisition de l'image, etc.) en dehors de l'application via requêtes http.

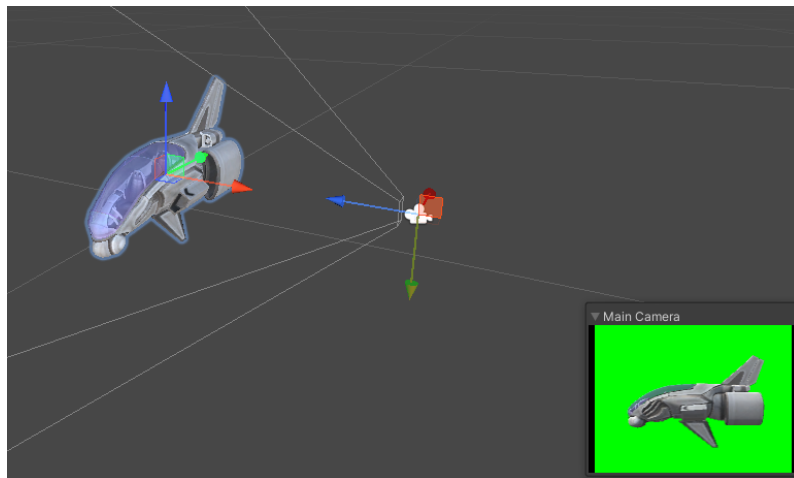


FIGURE 2 – Standalone Unity "AR_Engine"

- Une classe C++ "unityInterface" contenant différentes fonctions permettant de lancer des requêtes http depuis un programme C++ afin de contrôler la caméra virtuelle de l'application Unity.
- Les bibliothèques C++ pré-compilés nécessaires pour le développement du programme demandé :
 - ViSP et 2 dépendances (Eigen et OpenCV),
 - CURL pour l'interface avec le Standalone Unity.
- Un projet d'exemple "RA_Project" que vous pouvez utiliser comme base de travail. Ce programme vous montre comment :
 - ouvrir et initialiser la webcam usb,
 - acquérir et afficher une frame avec cette webcam,
 - se connecter au serveur http du Standalone Unity,
 - positionner la caméra virtuelle du Standalone Unity,
 - générer et récupérer un rendu de la caméra virtuelle du Standalone Unity.

3 Travail à réaliser

Rappel : connaissant les coordonnées 3D de points exprimées dans le repère d'un objet et les coordonnées 2D de leurs projections dans une image, il est alors possible d'estimer la pose de la caméra par rapport à cet objet.

3.1 Détection et suivi des caractéristiques visuelles

Afin de faire le lien entre le monde réel et le monde virtuel nous utiliserons le marqueur de la FIGURE 3.

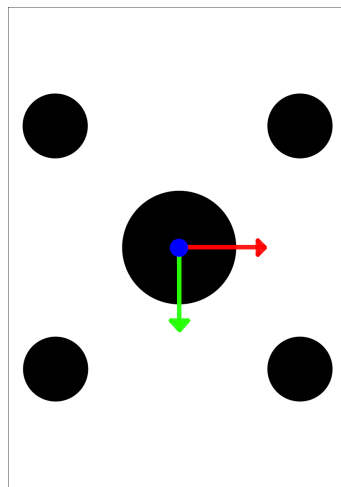


FIGURE 3 – Marqueur utilisé dans ce TP

Vous devez programmer la détection et le suivi des déplacements du centre des 4 petits cercles de ce marqueur dans les images acquises par la webcam. Pour plus de facilité, la détection de ces 4 cercles se fera manuellement en cliquant simplement dans l'image (voir la classe `vpDot`²).

3.2 Calcul de la pose de la caméra réelle

Comme étudié en cours, différentes approches sont possibles pour calculer la pose d'une caméra par rapport à un objet à partir de correspondances 2D/3D. Plusieurs de ces algorithmes basés points sont implémentés dans la classe `vpPose`³.

3.3 Génération du rendu virtuel

Lorsque la pose de votre webcam par rapport au marqueur est estimée, il vous est alors possible de modifier les paramètres extrinsèques de la caméra virtuelle du Standalone Unity (voir les fonctions de la classe "unityInterface") afin de positionner

2. <https://visp-doc.inria.fr/doxygen/visp-daily/classvpDot.html>

3. <https://visp-doc.inria.fr/doxygen/visp-daily/classvpPose.html>

et d'orienter cette dernière de la même telle que la pose de la caméra réelle mais vis-à-vis du vaisseau spatial. Puis, lorsque la caméra virtuelle est en place, vous pouvez alors générer et récupérer un rendu (voir les fonctions de la classe "unityInterface").

3.4 Composition de l'image augmentée

À ce stade, vous avez une image réelle et une image virtuelle qui devraient être "alignées". Il ne vous reste plus qu'à programmer une fonction d'incrustation de l'image virtuelle dans l'image réelle puis d'afficher le résultat.

4 Pour aller plus loin...

Paramétrer, tester, implémenter différents algorithmes de détection, suivi, calcul de pose afin de rendre votre application de Réalité Augmentée la plus robuste possible.